

# 基于改进代价计算和路径优化策略的匹配算法

周昊昊, 王晓旭, 王景隆, 赖康生\*

大连理工大学光电工程与仪器科学学院, 辽宁 大连 116024

**摘要** 为提升半全局匹配(SGM)算法效率,提出一种基于改进匹配代价计算和路径优化策略的立体匹配算法。代价计算阶段,通过对角线取点方式对局部二值模式(LBP)算子进行优化,降低时间复杂度和数据规模;代价聚合阶段,根据聚合逻辑选择5个方向进行扫描线优化,结合灰度相似性约束和距离约束条件,对聚合路径进行自适应权重赋值;再通过赢者通吃(WTA)策略计算初始视差值,通过左右一致性检测和二次多项式插值算法对视差图作进一步优化。最后算法在Middlebury 2.0和3.0数据平台上进行匹配效率验证,实验结果表明,所提算法相比SGM算法在不损失匹配准确度的情况下,代价计算阶段用时减少63.1%,代价聚合阶段用时减少39.3%,算法整体效率提升54.2%,达到效率提升的目的。

**关键词** 机器视觉;稠密匹配算法;改进局部二值模式算子;自适应权重;视差计算

中图分类号 TN911.73

文献标志码 A

DOI: 10.3788/LOP202259.1615006

## Matching Algorithm Based on Improved Cost Calculation and Path Optimization Strategy

Zhou Haohao, Wang Xiaoxu, Wang Jinglong, Lai Kangsheng\*

School of Optoelectronic Engineering and Instrumentation Science, Dalian University of Technology, Dalian 116024, Liaoning, China

**Abstract** To improve the efficiency of the semi-global matching (SGM) algorithm, a stereo matching algorithm based on better matching cost calculation and a path optimization strategy is proposed. In the cost calculation stage, the local binary patterns (LBP) operator was optimized by taking diagonal points to reduce time complexity and memory usage; in the cost aggregation stage, five directions were selected for scan line optimization according to the aggregation logic, combined with gray-scale similarity and distance constraint conditions to perform adaptive weight assignment; then the initial disparity value was calculated using the winner-takes-all (WTA) strategy, and the aggregation path was further optimized by the disparity map through left and right consistency detection and the quadratic polynomial interpolation algorithm. Finally, the algorithm's matching efficiency was validated using the Middlebury 2.0 and 3.0 data platforms. The experimental results show that when compared with the SGM algorithm, the proposed algorithm reduces the time used in the cost calculation stage by 63.1% and that in the cost aggregation stage by 39.3%. When the matching accuracy is slightly improved, the overall efficiency of the algorithm is increased by 54.2%, achieving the goal of efficiency improvement.

**Key words** machine vision; dense matching algorithm; improved local binary patterns operator; adaptive weight; disparity calculation

## 1 引言

图像匹配是双目立体视觉技术中的关键技术,通过计算图像对应点位置偏差构建视差图,可以进一步恢复物体深度关系<sup>[1-2]</sup>。伴随着三维信息应用场景越来越多,匹配算法逐渐开始关注匹配准确度和速度的协调性。解决这类问题,一方面通过优化算法结构,提

升匹配效率<sup>[3]</sup>,另一方面通过图形处理器(GPU)、现场可编程逻辑门阵列(FPGA)和数字信号处理器(DSP)等硬件设备<sup>[4]</sup>满足速度要求。本文研究内容针对改进立体匹配算法效率,以分阶段提升匹配速度为目的。Schartstein等<sup>[1]</sup>总结了当前传统匹配算法的基本思路,将立体匹配归纳为4个步骤:1)匹配代价计算;2)代价聚合;3)视差计算;4)视差后处理。

收稿日期: 2021-05-18; 修回日期: 2021-06-25; 录用日期: 2021-08-03

通信作者: laiksh@dlut.edu.cn

匹配代价计算阶段目的是实现像素描述及对应点间相似性度量<sup>[5-7]</sup>,完成由图像数据向匹配代价的转换。常用的代价表示方法有灰度差绝对值、灰度平方差、归一化互相关、秩变换以及像素序列化等。其中,局部二值模式(LBP)是一种典型的非参数变换方法,近年来较多学者提出了LBP算子的改进方案。宋巍等<sup>[8]</sup>提出增加内圈编码用于提高邻域信息利用率,提升了图像弱纹理和重复纹理区域匹配效果;程德强等<sup>[9]</sup>提出融合边缘信息进行LBP变换,以增强视差图中边缘质量;萧红等<sup>[10]</sup>提出通过加权平均作为变换参考值,结合图像梯度计算初始匹配代价。

代价聚合阶段目标是通过局部匹配代价传播,实现匹配代价空间的优化<sup>[4]</sup>。目前应用较多的有滤波策略<sup>[9-11]</sup>、支撑域策略<sup>[5,11]</sup>、扫描线优化<sup>[12-14]</sup>及最小生成树(MST)<sup>[7,15]</sup>等方法。其中,扫描线优化算法是动态规划算法的近似求解,将全局约束的马尔可夫随机场转变为若干方向累积求和,引入正则化约束条件以确保聚合代价的有效性,实现了代价空间中匹配异常点的修正目的。

为平衡匹配准确度与计算速度的关系,本文提出一种基于改进匹配代价计算和路径优化策略的立体匹

配算法。首先,通过对角线取点方式对LBP算法改进,优化了该阶段的时间复杂度和数据规模;然后根据聚合逻辑选择5个方向扫描线优化,结合灰度相似性和距离约束条件进行路径权重的自适应计算;最后,采用赢者通吃(WTA)策略进行视差计算,通过二次多项式插值和左右一致性检测算法对视差空间进一步优化,得到最终视差图。

## 2 算法描述

本文算法以经过核线校正后的双目像对为输入,经过匹配代价计算、代价聚合、视差计算及视差后处理等步骤得到最终视差,以视差图作为最终输出,算法总体流程如图1所示。匹配代价计算采用改进的LBP算子,通过对角线取点的方式降低计算复杂度;代价聚合首先根据计算逻辑顺序,将8路径聚合优化为5路径,然后结合自适应权重思想,根据灰度相似性约束和距离约束对聚合路径添加权重系数;视差计算采用WTA算法选择代价空间中最小代价对应索引作为初始视差值,得到初始视差图;视差后处理包括左右一致性检测、异常值填充、子像素优化等,获得最终视差图。

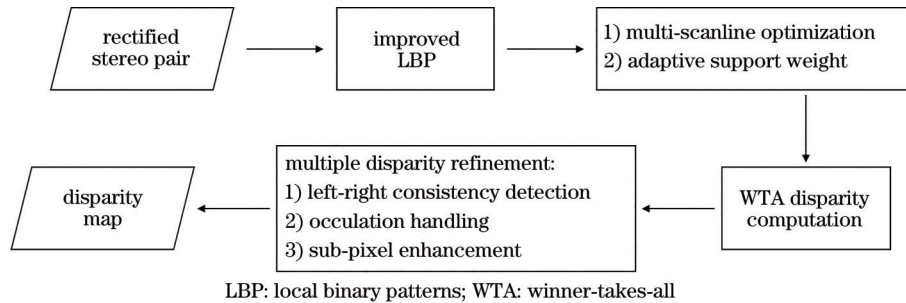


图1 提出算法流程图

Fig. 1 Diagram of proposed method

### 2.1 代价计算

匹配代价计算包含像素描述以及待匹配点之间相似性评价,完成由图像到代价空间的转换。代价空间通常被表示为 $W \times H \times D$ 的三维矩阵<sup>[15]</sup>,其中 $W$ 和 $H$ 分别表示源图像的宽度和高度, $D$ 表示像对视差选取范围,是由双目系统决定的参数。

LBP算子<sup>[16]</sup>定义 $n \times n$ 支撑窗口,以窗口中心像素灰度为参考值,将邻域像素灰度与参考值对比,通过二进制编码表示支撑窗口内灰度序列。公式描述为

$$C_{LBP}(x_c, y_c) = \sum_{n=0}^{n-1} 2^n \cdot \delta(i_n, i_c), \quad (1)$$

式中: $(x_c, y_c)$ 为窗口中心参考像素点坐标; $i_n$ 和 $i_c$ 分别表示邻域像素和参考像素灰度值; $n$ 表示邻域像素编号。 $\delta$ 函数定义为

$$\delta(i_n, i_c) = \begin{cases} 0, & \text{if } i_n \leq i_c \\ 1, & \text{if } i_n > i_c \end{cases} \quad (2)$$

传统LBP算子对像素的描述是通过将支撑窗口

内所有邻域像素转换二进制序列完成的,该过程并未考虑采样噪声对结果的影响。但在弱纹理区域,噪声的存在使得周围像素灰度在参考点灰度上下波动,此时 $\delta$ 函数结果是极易受到影响的。因此本文在代价计算阶段修改了感知窗口内像素采样规则,通过对角线取点方式降低噪声影响,同时可以精简掉大部分冗余像素信息。改进的代价计算原理如图2所示。

汉明(Hamming)距离变换通过统计两个比特串中不同比特位的个数衡量差异度,实现原理为对两个比特串进行异或运算,统计结果中1的个数。匹配代价 $C(u, v, d)$ 表示为

$$C(u, v, d) = \sum_{(u,v) \in \omega} \text{Hamming}[C_{LBP_1}(u, v) - C_{LBP_r}(u-d, v)], \quad (3)$$

式中, $C_{LBP_1}$ 和 $C_{LBP_r}$ 分别代表参考图像和目标匹配图像中的LBP比特串。

改进的LBP算子与传统LBP算子变换效果如图3所示。可以看出,改进的LBP算子在弱纹理区域表现

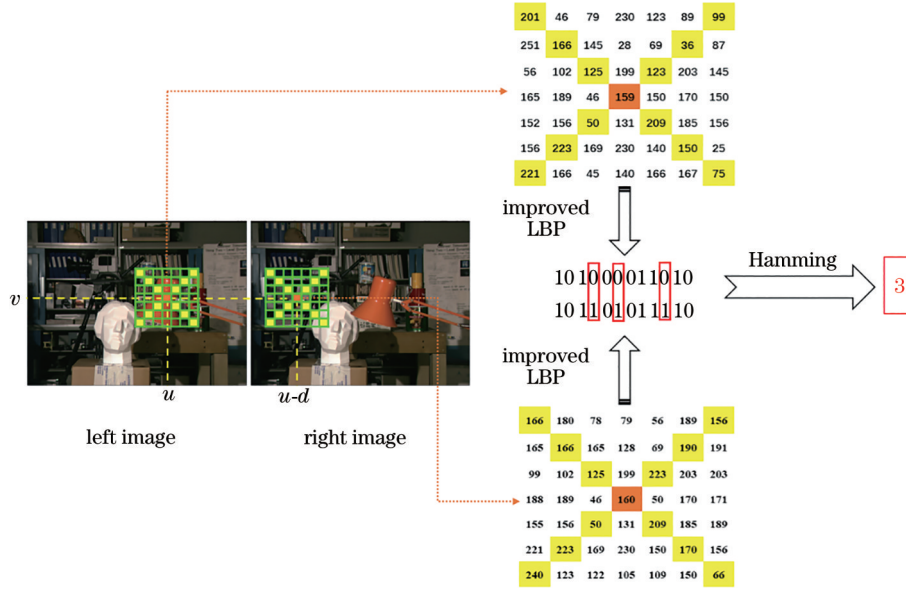


图 2 匹配代价计算过程

Fig. 2 Process of matching cost calculation

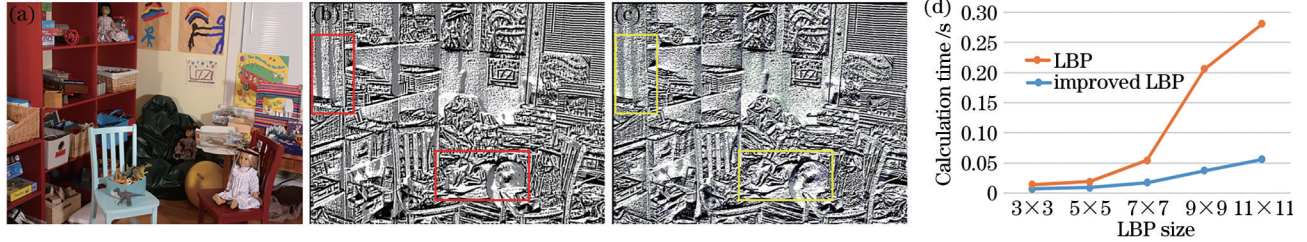


图 3 改进的 LBP 算子与原始 LBP 算子变换效果。(a)源图像;(b)原始 LBP 算子计算结果;(c)改进 LBP 算子计算结果;(d)计算时间对比

Fig. 3 Transform effects of improved LBP operator and initial LBP operator. (a) Source image; (b) result of initial LBP operator; (c) result of improved LBP operator; (d) calculation time comparison

效果更好,一定程度上抑制了冗余像素对代价计算结果的影响,在时间复杂度上实现了由  $O(n^2)$  到  $O(n)$  的优化,数据存储需求由  $n^2 - 1$  位比特减少为  $2(n - 1)$  位。

### 2.2 代价聚合

本文通过扫描线优化结合自适应权重算法进行代价聚合。扫描线优化算法<sup>[13]</sup>根据匹配点邻域视差关系,构造了包含惩罚项的全局能量函数  $E(D)$ , 匹配代价聚合就是实现能量函数最小化的过程。定义能量函数  $E(D)$  表示为

$$E(D) = \sum_p (C(p, D_p) + \sum_{q \in N_p} P_1 T[|D_p - D_q| = 1] + \sum_{q \in N_p} P_2 T[|D_p - D_q| > 1]), \quad (4)$$

$$T[\text{boolean}] = \begin{cases} 1, & \text{true} \\ 0, & \text{false} \end{cases}, \quad (5)$$

式中:  $D_p, D_q$  表示像素  $p, q$  点视差;  $P_1$  和  $P_2$  是惩罚系数;  $N_p$  表示聚合路径上经过的像素点。累加项中  $C(p, D_p)$  表示  $p$  点视差为  $D_p$  时对应的初始匹配代价; 第二项表示中心点与路径像素点如果视差相差 1, 累

加值为  $P_1$  的惩罚系数; 第三项表示中心点与路径像素视差相差大于 1, 累加值为  $P_2$  的惩罚系数。其中, 第二项和第三项也被称为平滑项, 通过平滑项的引入, 可以实现视差的邻域约束, 得到更加平滑的视差图。

能量函数最小化是一个非确定多项式(NP)完全问题<sup>[14]</sup>, 扫描线优化算法将该问题分解为不同路径的代价聚合过程:

$$L_r(p, d) = C(p, d) + \min \left\{ \begin{array}{l} L_r(p-r, d) \\ L_r(p-r, d-1) + P_1 \\ L_r(p-r, d+1) + P_1 \\ \min_i L_r(p-r, i) + P_2 \end{array} \right\} - \min_i L_r(p-r, i), \quad (6)$$

式中: 下标  $r$  表示聚合路径; 以从左向右聚合为例分析,  $L_r(p, d)$  表示在位置  $p$  点, 视差为  $d$  时, 按聚合路径  $r$  方向从中心点左侧逼近的聚合代价值: 第一项  $C(p, d)$  表示中心点初始匹配代价; 第二项表示对左侧点进行匹配代价修改后选择最小代价值; 在第三项中减去上一像素点最小代价, 目的是防止多路径聚合

累加后匹配代价值过大而失去代表性。

在算法实现过程中,多路径聚合并非同时运算,而是有一定逻辑次序。图 4 中  $\overrightarrow{(0,1)}$ 、 $\overrightarrow{(1,1)}$ 、 $\overrightarrow{(1,0)}$  和  $\overrightarrow{(-1,1)}$  4 个方向可以根据前面已完成代价聚合的数据进行传递,而其他 4 个方向是根据初始匹配代价数据参与运算。为了提高算法匹配效率,本文根据有效性原则将 8 路径聚合改为图 5 所示的 5 路径,保留  $\overrightarrow{(0,-1)}$  方向的目的是增强水平视差约束。

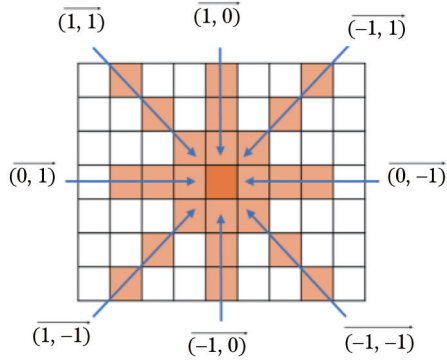


图 4 传统 8 路径代价聚合

Fig. 4 Traditional eight-path cost aggregation

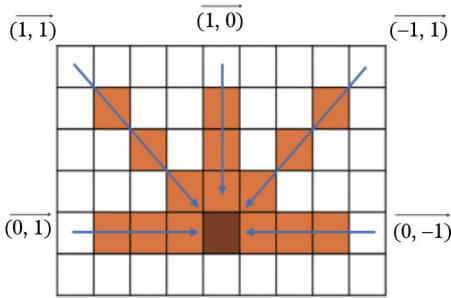


图 5 提出方法路径选择

Fig. 5 Path selection in proposed method

考虑到图像灰度变化剧烈的地方往往存在于物体边缘,在此位置容易发生深度不连续的现象。根据灰度相似性与距离约束原则,本文设计了自适应权重算法,并将其应用到聚合路径中:

$$W_r(p) = \sum_{i=0,1,2,\dots,R} W_{ir}(p - i \cdot r), \quad (7)$$

$$W_{ir}(p - i \cdot r) = \begin{cases} (R+1) - |i|, & \text{if } |I(p) - I(p - i \cdot r)| < \xi \\ 0, & \text{else} \end{cases}, \quad (8)$$

式中  $r$  表示聚合方向。本文算法聚合路径保留 5 个方向,即  $R=5$ 。本文假设相同物体表面灰度差不超过 10 个灰阶,设置灰度阈值参数  $\xi=10$ 。通过累加方式吸收多路径聚合代价:

$$S(p, d) = \sum_r W_r(p) \cdot L_r(p, d). \quad (9)$$

### 2.3 视差计算

代价聚合完成了代价空间的修正,通过  $S(p, d)$  表

示聚合后的匹配代价。在视差计算阶段,本文算法通过 WTA 策略<sup>[11]</sup>计算像素的初始视差。具体实现为:首先在代价集内根据像素坐标定位位置,然后搜索聚合代价最优的点作为对应的匹配点,取其对应的索引作为初始视差值。公式表示为

$$d_{\text{ini}}(p) = \underset{d \in D}{\operatorname{argmin}} [S(p, d)], \quad (10)$$

式中:  $D$  表示双目像对离散的视差集合;  $d_{\text{ini}}(p)$  表示像素点  $p$  的初始视差,通常选取最小代价表示最优匹配关系。经过该步骤后得到的初始视差图,可以直观反映本文算法改进的效果[图 6(d)],但仍然存在噪声和前景遮挡带来的误匹配,对细小结构和倾斜表面的表征也不够充分,需要在视差优化环节做进一步处理。

### 2.4 视差优化

视差优化环节包括左右一致性检测、遮挡/误匹配处理、子像素拟合等方式对视差图进一步优化。左右一致性检测可以表示为

$$|d_L(p) - d_R[p - d_L(p)]| \leq \zeta, \quad (11)$$

式中:  $d_L(p)$  为左视差图中像素  $p$  点的视差值;  $d_R[p - d_L(p)]$  表示该点在右视差图中对应点的视差值;设置阈值  $\zeta=2$ 。对满足阈值条件的左右视差通过取平均值作进一步修正,否则判定为异常点。

异常点出现原因可分为前景遮挡和误匹配。由于遮挡区域通常是前景物体造成的,因此通过背景视差传播进行遮挡点插值;对于误匹配区域填充根据灰度约束,选择邻域中灰度最相近的视差作为误匹配点的视差值。为了进一步提升视差精度,通过二次多项式插值进行子像素拟合:

$$d_{\text{sub}} = d_{\text{ini}} + \frac{c_1 - c_2}{2(c_1 + c_2 - 2c_0)}, \quad (12)$$

式中:  $d_{\text{sub}}$  表示子像素拟合后的视差值;  $d_{\text{ini}}$  是双边滤波后的像素  $p$  点的视差值;  $c_1$  和  $c_2$  分别是像素  $p$  在视差  $d_{\text{ini}} \pm 1$  时的聚合代价。子像素拟合可以有效阻止锯齿状视差边缘的产生,更加细致地描述深度渐进变化的斜面等区域。

## 3 实验结果与分析

本文算法使用 Middlebury 立体匹配平台<sup>[17-18]</sup>提供的数据进行实验,实验环境为: Visual studio 2017 下 C++ 编程语言, Windows10, X64 系统, Intel Core i7-8550U CPU, 8 G 内存。实验参数设置如表 1 所示。为定量比较本文算法与半全局匹配(SGM)算法的效果,两种算法代价计算窗口统一设为  $7 \times 7$ ; 参数  $P_1$  和  $P_2$  的选取参照了 Hirschmüller 等在文献[14]中的经验值;由于实验数据最大视差不超过 64,匹配结果图灰度统一放大  $\lambda$  倍。本文提出算法和 SGM 算法在 Middlebury 2.0 测试集中匹配效果如图 6 所示。

表 1 提出算法参数设置  
Table 1 Parameters of proposed algorithm

Parameter	LBP size	$P_1$	$P_2$	$R$	$\xi$	$\zeta$	$\lambda$
Value	$7 \times 7$	10	150	5	10	2	5

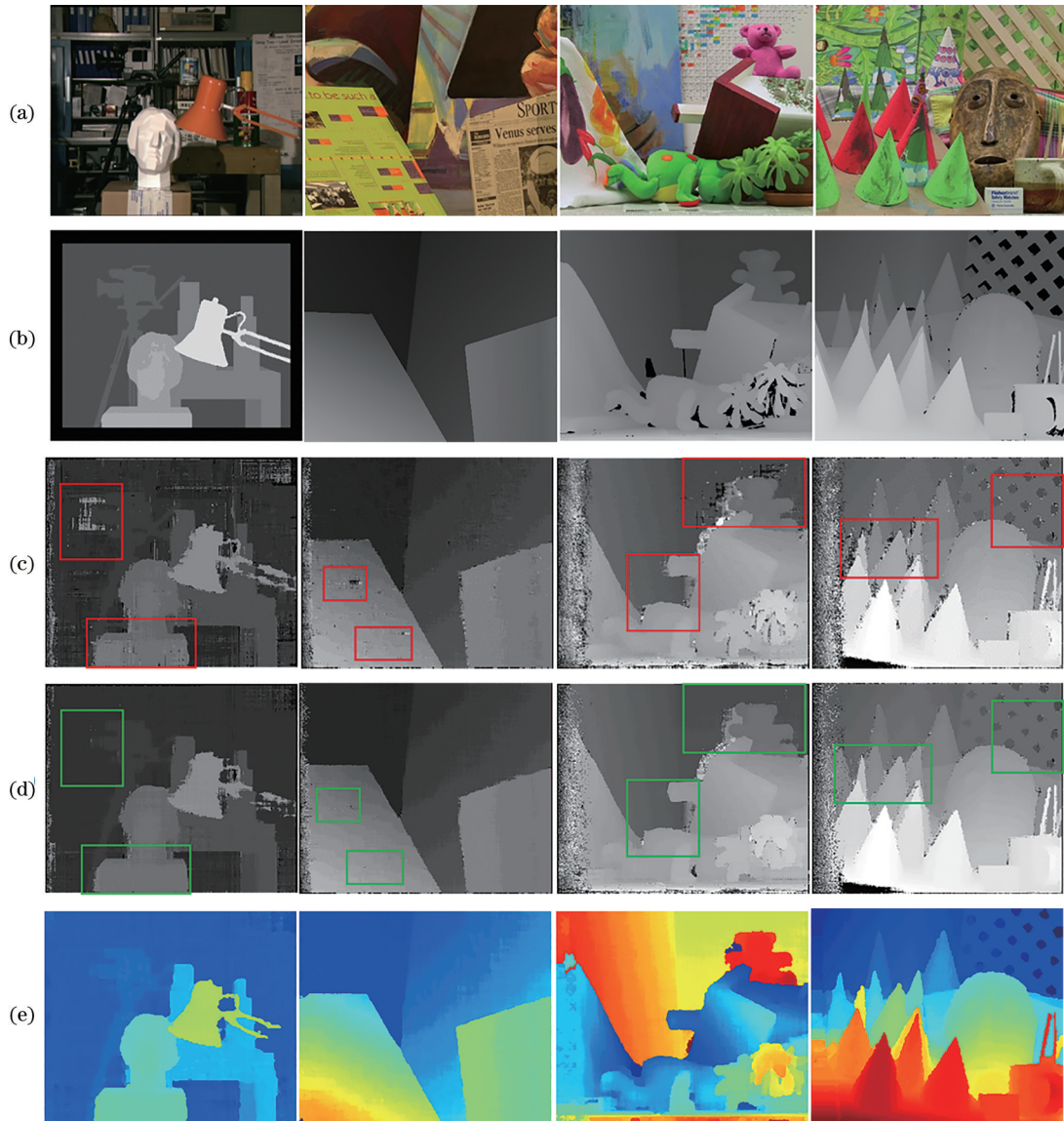


图 6 实验效果图。(a)原始左视图(tsukuba、venus、teddy、cones);(b)真实视差图;(c) SGM匹配视差图;(d)提出算法匹配视差图;  
(e)提出算法伪彩色视差图

Fig. 6 Experimental effect images. (a) Raw images in left view (tsukuba, venus, teddy, and cones); (b) true disparity map; (c) disparity map of SGM; (d) disparity map of proposed algorithm; (e) disparity map in pseudo-color of proposed algorithm

从图 6(c)中可以看出, SGM 算法本身匹配效果已经不错,这也是本文提出在该算法基础上进行优化的原因。SGM 算法误匹配多发生在弱纹理和重复纹理等区域,视差图中存在较多突变点。提出算法在代价聚合阶段引入了自适应权重,增强了灰度约束和距离约束,弥补了原算法的缺陷,对边缘保持也体现出积极的作用。计算时间优化是提出算法的重点。在 debug 和 release 两种编译模式下,提出算法与 SGM 算法的计算时间对比如表 2 和表 3 所示。

从表 2 和表 3 可以看出,代价计算阶段,两种编译

模式下提出算法相比 SGM 算法平均计算时间分别减少了 67.3% 和 58.8%,与该阶段时间复杂度的优化目标相符;在代价聚合阶段,平均计算时间分别减少了 38.4% 和 40.2%,表示自适应权重算法并没有占用太多计算时间,这是因为提出算法简化了 ASW 计算,并不涉及指数运算等复杂过程。

当阈值为 1 时,提出算法与其他算法<sup>[19-22]</sup>的匹配准确率实验对比如表 4 所示,其中 nonocc、all、disc 分别表示场景图片的非遮挡区域、全部区域、视差不连续区域。可以看出,在 Middlebury 2.0 数据集中提出算法平均误

表 2 debug 模式下计算时间对比  
Table 2 Calculation time comparison in debug mode

unit: s

Algorithm	Tsukuba		Venus		Teddy		Cones	
	$d \in [0, 16]$		$d \in [0, 20]$		$d \in [0, 64]$		$d \in [0, 64]$	
	Cost calculation	Cost aggregation	Cost calculation	Cost aggregation	Cost calculation	Cost aggregation	Cost calculation	Cost aggregation
SGM	0.170	8.472	0.280	15.777	0.695	49.987	0.713	50.114
Proposed	0.053	5.354	0.074	10.076	0.253	29.979	0.254	29.714
<b>Improved time</b>	<b>0.117</b> (68.0%)	<b>3.118</b> (36.8%)	<b>0.206</b> (73.5%)	<b>5.701</b> (36.1%)	<b>0.442</b> (63.5%)	<b>20.008</b> (40.0%)	<b>0.459</b> (64.3%)	<b>20.400</b> (40.7%)

表 3 release 模式下计算时间对比  
Table 3 Calculation time comparison in release mode

unit: s

Algorithm	Tsukuba		Venus		Teddy		Cones	
	$d \in [0, 16]$		$d \in [0, 20]$		$d \in [0, 64]$		$d \in [0, 64]$	
	Cost calculation	Cost aggregation	Cost calculation	Cost aggregation	Cost calculation	Cost aggregation	Cost calculation	Cost aggregation
SGM	0.037	0.088	0.071	0.163	0.196	0.445	0.205	0.493
Proposed	0.017	0.045	0.025	0.097	0.075	0.311	0.093	0.289
<b>Improved time</b>	<b>0.020</b> (54.0%)	<b>0.043</b> (48.8%)	<b>0.046</b> (64.7%)	<b>0.066</b> (40.5%)	<b>0.121</b> (61.7%)	<b>0.134</b> (30.1%)	<b>0.112</b> (54.6%)	<b>0.204</b> (41.3%)

表 4 阈值为 1 时误匹配率对比  
Table 4 Comparison of mismatched ratio with threshold is 1

unit: %

Algorithm	Tsukuba			Venus			Teddy			Cones			Average
	nonocc	all	disc	nonocc	all	disc	nonocc	all	disc	nonocc	all	disc	
RTcensus <sup>[19]</sup>	5.1	6.3	19.2	1.6	2.4	14.2	8.0	13.8	20.3	4.1	9.5	12.2	9.73
ADcensus <sup>[20]</sup>	1.1	1.5	5.7	0.1	0.3	1.2	4.1	6.2	10.9	2.4	7.3	7.0	3.97
C-SemiGlob <sup>[23]</sup>	2.6	3.3	9.9	0.3	0.6	3.2	5.1	11.8	13.0	2.8	8.4	8.2	5.76
planeFitSGM <sup>[21]</sup>	3.1	4.2	14.9	1.1	1.9	14.6	5.7	11.6	17.1	3.8	9.3	11.3	8.21
SGM <sup>[14]</sup>	3.3	4.0	12.8	1.0	1.6	11.3	6.0	12.2	16.3	3.1	9.8	8.9	7.51
SGMDDW <sup>[22]</sup>	2.3	4.4	11.8	1.2	2.7	16.8	6.5	14.5	17.5	5.6	14.2	14.8	9.36
<b>Proposed</b>	<b>3.3</b>	<b>3.5</b>	<b>12.7</b>	<b>0.9</b>	<b>1.1</b>	<b>9.8</b>	<b>7.3</b>	<b>9.1</b>	<b>14.4</b>	<b>3.9</b>	<b>7.3</b>	<b>11.7</b>	<b>7.08</b>

匹配率为 7.08%，低于 RTcensus、planeFitSGM、SGMDDW、SGM 等算法，略高于 ADcensus 及 C-SemiGlob 算法，但提出算法平均计算时间为 0.488 s，相较 ADcensus 算法的 15.570 s 和 C-SemiGlob 算法的 3.110 s 有明显优势。

当阈值为 0.5 时，其他参数设置与表 1 完全相同，所提算法与其他算法实验结果对比如表 5 所示。可以看出，提出算法在小阈值条件下整体仍然有较好的匹配效果，在视差不连续区域略有劣化，反映出提出算法在视差修正环节需要继续提升。

表 5 阈值为 0.5 时误匹配率对比  
Table 5 Comparison of mismatched ratio when threshold is 0.5

unit: %

Algorithm	Tsukuba			Venus			Teddy			Cones			Average
	nonocc	all	disc	nonocc	all	disc	nonocc	all	disc	nonocc	all	disc	
RTcensus <sup>[19]</sup>	12.9	14.1	28.1	3.7	4.6	17.8	11.4	18.6	27.7	5.5	11.8	15.9	14.34
ADcensus <sup>[20]</sup>	26.8	27.0	21.1	4.1	4.6	8.0	10.6	13.8	20.1	6.6	12.4	11.9	13.92
C-SemiGlob <sup>[23]</sup>	13.9	14.7	18.9	3.3	3.8	10.9	9.8	17.4	22.8	5.4	11.7	12.8	12.12
planeFitSGM <sup>[21]</sup>	9.2	10.4	23.3	2.3	3.2	15.9	9.0	17.0	25.5	5.2	11.6	14.7	12.28
SGM <sup>[14]</sup>	13.4	14.3	20.3	4.6	5.4	15.7	11.0	18.5	26.1	4.9	12.5	13.5	13.35
SGMDDW <sup>[22]</sup>	15.5	17.5	23.1	8.1	9.6	25.6	15.7	23.9	31.5	14.5	22.8	26.0	19.48
<b>Proposed</b>	<b>6.3</b>	<b>8.9</b>	<b>20.1</b>	<b>5.5</b>	<b>5.9</b>	<b>17.3</b>	<b>8.2</b>	<b>10.0</b>	<b>30.5</b>	<b>8.5</b>	<b>10.1</b>	<b>17.4</b>	<b>12.39</b>

为进一步验证提出算法的有效性和鲁棒性,将该算法在 Middlebury 3.0 数据集中全部区域进行准确率验证,提出算法与其他算法<sup>[13-14,23-26]</sup>对比结果如表 6 所示。

表 6 全部区域误匹配率对比  
Table 6 Comparison of mismatched ratio in all regions

unit: %

Stereo pairs	Mismatched ratio						
	MDP <sup>[23]</sup>	SRM <sup>[24]</sup>	Cens5 <sup>[25]</sup>	SGBM1 <sup>[13]</sup>	SGM <sup>[14]</sup>	DTS <sup>[26]</sup>	Proposed
Adiron	25.0	28.7	37.0	39.5	29.1	20.9	25.5
ArtL	15.1	15.8	17.5	19.0	11.5	17.7	11.7
Jadepl	35.9	38.2	35.9	36.0	28.1	34.3	20.2
Motor	30.4	27.6	32.5	25.9	25.5	21.1	17.9
MotorE	28.7	30.3	28.9	36.9	22.5	21.8	15.8
Piano	33.6	38.1	33.6	36.6	26.1	33.0	22.3
PianoL	41.7	47.9	47.4	58.6	42.1	39.0	39.5
Pipes	29.0	23.2	23.6	20.4	20.5	24.9	14.4
Playrm	42.7	44.6	47.4	45.0	38.3	40.6	26.8
Playt	47.8	35.0	67.9	52.4	64.7	34.7	45.3
PlaytP	28.8	32.9	29.1	29.2	24.7	32.2	17.3
Recyc	31.3	32.4	33.6	33.0	27.0	24.1	18.9
Shelvs	54.5	54.4	62.1	57.2	59.0	46.9	41.3
Teddy	10.8	10.6	12.3	18.1	10.1	10.3	8.2
Vintge	43.6	49.5	61.2	55.0	51.3	47.6	35.9
Austr	36.3	29.3	63.2	51.8	58.1	24.5	40.7
AustrP	18.4	27.2	19.0	24.1	10.2	23.6	17.1
Bicyc2	24.4	25.6	19.2	21.4	15.5	17.5	17.9
Class	30.0	29.5	33.7	24.4	30.4	21.7	31.3
ClassE	48.8	40.8	57.3	78.5	54.6	38.3	38.2
Compu	16.9	18.9	30.0	20.8	21.2	20.9	17.9
Crusa	40.7	45.5	61.3	47.8	48.8	44.3	34.2
CrusaP	32.8	45.2	47.7	41.9	27.9	45.0	19.5
Djemb	20.9	21.8	19.2	13.4	13.9	19.2	23.7
DjembL	47.3	42.5	49.7	63.5	44.1	33.8	30.9
Hoops	41.2	41.5	51.2	49.8	44.2	39.3	31.9
Livgrm	28.8	33.4	39.5	34.6	33.1	29.1	23.2
Nkuba	38.3	38.7	36.4	34.6	32.4	36.1	20.7
Plants	33.5	32.3	43.0	35.4	31.9	31.7	22.3
Stairs	35.2	42.5	55.0	59.2	44.7	29.8	31.3
Average	30.9	32.4	36.6	35.2	29.7	28.6	25.4

从表 6 中可以看出,提出算法在 Middlebury 3.0 数据集中平均误匹配率 25.4%, 低于 SGM 算法 4 个百分点。在匹配速度方面, release 编译模式下提出算法平均计算时间 0.47 s, 优于 SGM 算法的 1.17 s, 整体效率提升 59.8%。该结果与 Middlebury 2.0 数据集中测试结果基本一致。

## 4 结 论

提出了一种基于改进代价计算和聚合策略的立体匹配算法。该算法在 SGM 算法基础上进行改进, 目标是在匹配准确率不损失的前提下提升匹配算法效

率。首先在匹配代价计算阶段, 通过对角线取点方式对 LBP 算子进行精简, 实现计算复杂度和数据规模的优化。然后在代价聚合阶段, 根据代码实现过程改为 5 个方向扫描线优化, 同时将改进的自适应权重结合到代价传递过程中。通过 WTA 算法计算初始视差值, 结合左右一致性检测和二次多项式插值算法对视差图进一步细化。最后将该算法在 Middlebury 2.0 和 3.0 数据平台上进行匹配效率验证, 实验结果表明, 该算法相比 SGM 算法在匹配准确率略微提升的情况下, 代价计算阶段平均用时减少 63.1%, 代价聚合阶段平均用时减少 39.3%, 算法整体效率提升 54.2%,

达到算法效率提升的目的。

在实际算法验证过程中,匹配效率的另一个制约因素是计算机的磁盘读取性能和内存分配。因此今后研究将在硬件设备上做进一步优化和验证,以提升算法的效率和鲁棒性。

### 参 考 文 献

- [1] Scharstein D, Szeliski R, Zabih R. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms[C]//Proceedings IEEE Workshop on Stereo and Multi-Baseline Vision, December 9-10, 2001, Kauai, HI, USA. New York: IEEE Press, 2001: 131-140.
- [2] Hirschmuller H, Scharstein D. Evaluation of stereo matching costs on images with radiometric differences[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2009, 31(9): 1582-1599.
- [3] Hamzah R A, Ibrahim H. Literature survey on stereo vision disparity map algorithms[J]. Journal of Sensors, 2016, 2016: 8742920.
- [4] Li Y, Li Z W, Yang C, et al. High throughput hardware architecture for accurate semi-global matching[J]. Integration, 2019, 65: 417-427.
- [5] 闫利, 王芮, 刘华, 等. 基于改进代价计算和自适应引导滤波的立体匹配[J]. 光学学报, 2018, 38(11): 1115007.  
Yan L, Wang R, Liu H, et al. Stereo matching method based on improved cost computation and adaptive guided filter[J]. Acta Optica Sinica, 2018, 38(11): 1115007.
- [6] Yoon K J, Kweon I S. Adaptive support-weight approach for correspondence search[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2006, 28(4): 650-656.
- [7] Yang Q X. A non-local cost aggregation method for stereo matching[C]//2012 IEEE Conference on Computer Vision and Pattern Recognition, June 16-21, 2012, Providence, RI, USA. New York: IEEE Press, 2012: 1402-1409.
- [8] 宋巍, 魏新宇, 张明华, 等. 基于改进代价计算和视差候选策略的立体匹配[J]. 激光与光电子学进展, 2021, 58(2): 0215001.  
Song W, Wei X Y, Zhang M H, et al. Stereo matching based on improved cost calculation and a disparity candidate strategy[J]. Laser & Optoelectronics Progress, 2021, 58(2): 0215001.
- [9] 程德强, 庄焕东, 于文洁, 等. 基于边缘加权的跨尺度局部立体匹配算法[J]. 激光与光电子学进展, 2019, 56(21): 211504.  
Cheng D Q, Zhuang H D, Yu W J, et al. Cross-scale local stereo matching based on edge weighting[J]. Laser & Optoelectronics Progress, 2019, 56(21): 211504.
- [10] 萧红, 田川, 张毅, 等. 基于改进 Census 变换与梯度融合的立体匹配算法[J]. 激光与光电子学进展, 2021, 58(2): 0215008.  
Xiao H, Tian C, Zhang Y, et al. Stereo matching algorithm based on improved census transform and gradient fusion[J]. Laser & Optoelectronics Progress, 2021, 58(2): 0215008.
- [11] 孔令寅, 朱江平, 应三丛. 基于引导图像和自适应支持域的立体匹配[J]. 光学学报, 2020, 40(9): 0915001.  
Kong L Y, Zhu J P, Ying S C. Stereo matching based on guidance image and adaptive support region[J]. Acta Optica Sinica, 2020, 40(9): 0915001.
- [12] Hirschmuller H. Accurate and efficient stereo processing by semi-global matching and mutual information[C]//2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, June 20-25, 2005, San Diego, CA, USA. New York: IEEE Press, 2005: 807-814.
- [13] Li Y S, Zheng S Y, Wang X N, et al. An efficient photogrammetric stereo matching method for high-resolution images[J]. Computers & Geosciences, 2016, 97: 58-66.
- [14] Hirschmüller H. Stereo processing by semiglobal matching and mutual information[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2008, 30(2): 328-341.
- [15] 马瑞青, 廖宁放, 强彦, 等. 同色异谱在二维模拟场景下对颜色恒常性的影响[J]. 光学学报, 2021, 41(18): 1833001.  
Ma R Q, Liao N F, Qiang Y, et al. Effect of metamer mismatching on color constancy in two-dimensional simulated scenes[J]. Acta Optica Sinica, 2021, 41(18): 1833001.
- [16] Zhu W. OpenCV image processing programming example [M]. Beijing: Publishing House of Electronics Industry, 2016: 312-313.
- [17] Baker S, Scharstein D, Lewis J P, et al. A database and evaluation methodology for optical flow[J]. International Journal of Computer Vision, 2011, 92(1): 1-31.
- [18] Scharstein D, Hirschmüller H. High-resolution stereo datasets with subpixel-accurate ground truth[C]//German Conference on Pattern Recognition (GCPR), September 2-5, 2014, Münster, Germany: Springer International Publishing, 2014: 31-42.
- [19] Humenberger M, Zinner C, Weber M, et al. A fast stereo matching algorithm suitable for embedded real-time systems[J]. Computer Vision and Image Understanding, 2010, 114(11): 1180-1202.
- [20] Mei X, Sun X, Zhou M C, et al. On building an accurate stereo matching system on graphics hardware[C]//2011 IEEE International Conference on Computer Vision Workshops, November 6-13, 2011, Barcelona, Spain. New York: IEEE Press, 2011: 467-474.
- [21] Humenberger M, Engelke T, Kubinger W. A census-based stereo vision algorithm using modified Semi-Global Matching and plane fitting to improve matching quality [C]//2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops, June 13-18, 2010, San Francisco, CA, USA. New York: IEEE Press, 2010: 77-84.
- [22] Michael M, Salmen J, Stallkamp J, et al. Real-time stereo vision: optimizing semi-global matching[C]//2013 IEEE Intelligent Vehicles Symposium, June 23-26, 2013, Gold Coast, QLD, Australia. New York: IEEE



- Press, 2013: 1197-1202.
- [23] Hirschmuller H. Stereo vision in structured environments by consistent semi-global matching[C]//2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, June 17-22, 2006, New York, NY, USA. New York: IEEE Press, 2006: 2386-2393.
- [24] Okae J, Du J, Hu Y M. Robust statistical approach to stereo disparity maps denoising and refinement[J]. Control Theory and Technology, 2020, 18(4): 348-361.
- [25] Hirschmüller H, Innocent P R, Garibaldi J. Real-time correlation-based stereo vision with reduced border errors [J]. International Journal of Computer Vision, 2002, 47 (1/2/3): 229-246.
- [26] Bapat A, Frahm J M. The domain transform solver[C]// 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 15-20, 2019, Long Beach, CA, USA. New York: IEEE Press, 2019: 6007-6016.